**Министерство науки и высшего образования Российской Федерации**

**федеральное государственное автономное образовательное учреждение высшего образования**

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**УНИВЕРСИТЕТ ИТМО»**

**ФАКУЛЬТЕТ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**ОТЧЕТ**

**по Лабораторной работе № 16**

**«Программирование задач с использованием структур в функциях, работа с файлами и структурами»**

Специальность 09.02.07«Информационные системы и программирование»

Дисциплина «Основы алгоритмов и программирование»

|  |  |
| --- | --- |
| Преподаватель:  Яковлева Н.М.  «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_202\_\_г.  Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Выполнил:  студент группы Y2231  Петров Н.В. |

Санкт-Петербург

2019/2020

Разработка структуры входных данных и выходных форм

В данном программном коде используются 2 основные структуры данных:

Структура DB, состоящая из указания кол-ва элементов типа Item, список которых – второе поле структуры DB.

Структура Item – один товар, поля структуры: артикул (код товара), наименование, количество и цена.

В качестве формата таблицы используется формат CSV.

Функциональные требования к программе

Программа считывает файл, записанный в формате CSV-таблицы, и преобразовывает каждую строку в структуру, описывающую один товар. В итоге получается массив структур. Далее программа выдаёт следующую информацию:

Проблемная часть:

1. Вывод наличия и стоимости изделия артикула Х (вводится с терминала);
2. Вывод ассортиментного списка детского трикотажа с указанием наименования, если известно, что артикул начинается с буквы W для дамского трикотажа, M – для мужского, C – для детского

Служебные функции:

1. Подпрограмма создания файла.
2. Подпрограмма просмотра файла
3. Подпрограмма добавления данных.
4. Подпрограмма редактирования данных.
5. Подпрограмма удаления данных.

Алгоритм проблемной части



Выбор программного обеспечения

Выбранная среда разработки – CLion.

Преимущества:

* Поддержка компиляции в WSL
* Статический анализатор кода с подсветкой синтаксиса
* Интеграция с Git/GitHub и другими VCS
* Человеческий дебагер
* Темная тема
* Поддержка копирования кода в Word с сохранением форматирования
* Поддержка экспорта кода в HTML
* Авто-рефактор кода
* Поддержка кучи плагинов

Программный код

main.cpp

#include <iostream>  
#include "module.h"  
  
using namespace std;  
  
  
int main() {  
 auto db = openDB("../db.csv");  
  
 int work\_mode;  
 cout << "Выберети режим работы:\n"  
 "1. Вывести одежду с требуемым кодом\n"  
 "2. Вывести детскую одежду" << endl;  
 cin >> work\_mode;  
  
 if (work\_mode == 1) {  
 char code[CODE\_SIZE];  
 cout << "Введите артикул: " << endl;  
 cin >> code;  
 print\_by\_code(code, db);  
 }  
 if (work\_mode == 2) print\_c\_code(db);  
  
 return 0;  
}

module.h

#ifndef LAB16\_\_\_MODULE\_H  
#define LAB16\_\_\_MODULE\_H  
  
#define CODE\_SIZE 4  
#define NAME\_SIZE 32  
#define DB\_SIZE 65535  
  
struct Item {  
 char code[CODE\_SIZE];  
 char name[NAME\_SIZE];  
 int amount;  
 int price;  
};  
  
struct DB {  
 int len;  
 Item\* table;  
};  
  
DB openDB(const char \*f\_name);  
void print\_by\_code(const char \*str, DB data);  
void print\_c\_code(DB data);  
void create\_DB(const char \*f\_name, DB data);  
void explore\_DB(const char \*f\_name);  
void clear\_DB(const char \*f\_name);  
DB add\_to\_DB(Item new\_raw, DB data);  
DB edit\_DB(int raw, Item new\_item, DB data);  
  
#endif //LAB16\_\_\_MODULE\_H

module1.cpp

#include <iostream>  
#include <cstring>  
#include "module.h"  
  
using namespace std;  
  
Item to\_item(const char \*str) {  
 Item out;  
 int text\_index = 0;  
 memset(out.code, '\0', CODE\_SIZE);  
 for (; str[text\_index] != ',' && text\_index < CODE\_SIZE; text\_index++) {  
 out.code[text\_index] = str[text\_index];  
 }  
 memset(out.name, '\0', NAME\_SIZE);  
 int local\_index;  
 for (local\_index = 0, text\_index++;  
 str[text\_index] != ',' && local\_index < NAME\_SIZE; text\_index++, local\_index++) {  
 out.name[local\_index] = str[text\_index];  
 }  
 char amount\_str[10] = {'\0'};  
 for (local\_index = 0, text\_index++; str[text\_index] != ',' && local\_index < 10; text\_index++, local\_index++) {  
 amount\_str[local\_index] = str[text\_index];  
 }  
 out.amount = atoi(amount\_str);  
 char price\_str[10] = {'\0'};  
 for (local\_index = 0, text\_index++; str[text\_index] != ',' && local\_index < 10; text\_index++, local\_index++) {  
 price\_str[local\_index] = str[text\_index];  
 }  
 out.price = atoi(price\_str);  
 return out;  
}  
  
DB openDB(const char \*f\_name) {  
 FILE \*fp;  
 if ((fp = fopen(f\_name, "r")) == nullptr) {  
 puts("Невозможно открыть файл");  
 exit(-1);  
 }  
  
 int len = 0;  
 char buff;  
 char text[DB\_SIZE];  
 int text\_index = 0;  
 while ((buff = getc(fp)) != EOF) {  
 if (buff == '\n') len++;  
 text[text\_index++] = buff;  
 }  
 fclose(fp);  
  
 Item db[len];  
 char \*text\_ptr = text;  
 for (int i = 0; i < len; i++) {  
 db[i] = to\_item(text\_ptr);  
 text\_ptr = strchr(text\_ptr, '\n') + 1;  
 }  
 return DB{len, db};  
}

module2.cpp

#include <iostream>  
#include <iomanip>  
#include <cstring>  
#include "module.h"  
  
using namespace std;  
  
void print\_by\_code(const char \*code, DB data) {  
 auto[len, db] = data;  
 cout << "┌" << string(CODE\_SIZE, '-') << "┬" << string(NAME\_SIZE, '-') << "┬" << string(10, '-') << "┬"  
 << string(10, '-') << "┐" << endl;  
 cout << "|" << left << setw(CODE\_SIZE) << "Code" << "|" << setw(NAME\_SIZE) << "Name" << "|" << setw(10) << "Amount"  
 << "|" << setw(10) << "Price" << "|" << endl;  
 cout << "├" << string(CODE\_SIZE, '-') << "┼" << string(NAME\_SIZE, '-') << "┼" << string(10, '-') << "┼"  
 << string(10, '-') << "┤" << endl;  
 for (int i = 0; i < len; i++) {  
 if (strcmp(code, db[i].code) == 0) {  
 cout << "|" << left << setw(CODE\_SIZE) << db[i].code << "|" << setw(NAME\_SIZE) << db[i].name << "|" << right  
 << setw(10) << db[i].amount << "|" << setw(10) << db[i].price << "|" << endl;  
 }  
 }  
 cout << "└" << string(CODE\_SIZE, '-') << "┴" << string(NAME\_SIZE, '-') << "┴" << string(10, '-') << "┴"  
 << string(10, '-') << "┘" << endl;  
}

module3.cpp

#include <iostream>  
#include <iomanip>  
#include <cstring>  
#include "module.h"  
  
using namespace std;  
  
void print\_c\_code(DB data) {  
 auto[len, db] = data;  
 cout << "Детская одежда" << endl;  
 cout << "┌" << string(CODE\_SIZE, '-') << "┬" << string(NAME\_SIZE, '-') << "┬" << string(10, '-') << "┬"  
 << string(10, '-') << "┐" << endl;  
 cout << "|" << left << setw(CODE\_SIZE) << "Code" << "|" << setw(NAME\_SIZE) << "Name" << "|" << setw(10) << "Amount"  
 << "|" << setw(10) << "Price" << "|" << endl;  
 cout << "├" << string(CODE\_SIZE, '-') << "┼" << string(NAME\_SIZE, '-') << "┼" << string(10, '-') << "┼"  
 << string(10, '-') << "┤" << endl;  
 for (int i = 0; i < len; i++) {  
 if (strncmp("C", db[i].code, 1) == 0) {  
 cout << "|" << left << setw(CODE\_SIZE) << db[i].code << "|" << setw(NAME\_SIZE) << db[i].name << "|" << right  
 << setw(10) << db[i].amount << "|" << setw(10) << db[i].price << "|" << endl;  
 }  
 }  
 cout << "└" << string(CODE\_SIZE, '-') << "┴" << string(NAME\_SIZE, '-') << "┴" << string(10, '-') << "┴"  
 << string(10, '-') << "┘" << endl;  
}

module4.cpp

#include <iostream>  
#include "module.h"  
  
using namespace std;  
  
void create\_DB(const char \*f\_name, DB data) {  
 FILE \* fp;  
 fp = fopen (f\_name,"wt");  
  
 for(int i = 0; i < data.len; i++){  
 Item\* raw = &data.table[i];  
 fprintf (fp, "%s,%s,%d,%d\n", raw->code, raw->name, raw->amount, raw->price);  
 }  
 fclose (fp);  
}

module5.cpp

#include <iostream>  
#include <iomanip>  
#include "module.h"  
  
using namespace std;  
  
void explore\_DB(const char \*f\_name) {  
 auto[len, db] = openDB(f\_name);  
 cout << "┌" << string(CODE\_SIZE, '-') << "┬" << string(NAME\_SIZE, '-') << "┬" << string(10, '-') << "┬"  
 << string(10, '-') << "┐" << endl;  
 cout << "|" << left << setw(CODE\_SIZE) << "Code" << "|" << setw(NAME\_SIZE) << "Name" << "|" << setw(10) << "Amount"  
 << "|" << setw(10) << "Price" << "|" << endl;  
 cout << "├" << string(CODE\_SIZE, '-') << "┼" << string(NAME\_SIZE, '-') << "┼" << string(10, '-') << "┼"  
 << string(10, '-') << "┤" << endl;  
 for (int i = 0; i < len; i++) {  
 cout << "|" << left << setw(CODE\_SIZE) << db[i].code << "|" << setw(NAME\_SIZE) << db[i].name << "|" << right  
 << setw(10) << db[i].amount << "|" << setw(10) << db[i].price << "|" << endl;  
 }  
 cout << "└" << string(CODE\_SIZE, '-') << "┴" << string(NAME\_SIZE, '-') << "┴" << string(10, '-') << "┴"  
 << string(10, '-') << "┘" << endl;  
}

module6.cpp

#include "module.h"  
  
using namespace std;  
  
bool str\_compare(const char \*str1, const char \*str2, int char\_size) {  
 for(int i = 0; i < char\_size; i++) {  
 if ((int)str1[i] < (int)str2[i]) return true;  
 }  
 return false;  
}  
  
DB add\_to\_DB(Item new\_raw, DB data) {  
 auto[len, db] = data;  
 Item new\_db[len+1];  
 int i;  
 for(i = 0; i < len+1; i++) {  
 if (str\_compare(db[i].code, new\_raw.code, CODE\_SIZE)) {  
 break;  
 }  
 new\_db[i] = db[i];  
 }  
 new\_db[i] = new\_raw;  
 for(; i < len+1; i++) {  
 new\_db[i+1] = db[i];  
 }  
 return DB{len, new\_db};  
}

module7.cpp

#include <iostream>  
#include "module.h"  
  
using namespace std;  
  
void clear\_DB(const char \*f\_name) {  
 FILE \* fp;  
 fp = fopen (f\_name,"w");  
 fclose(fp);  
}

module8.cpp

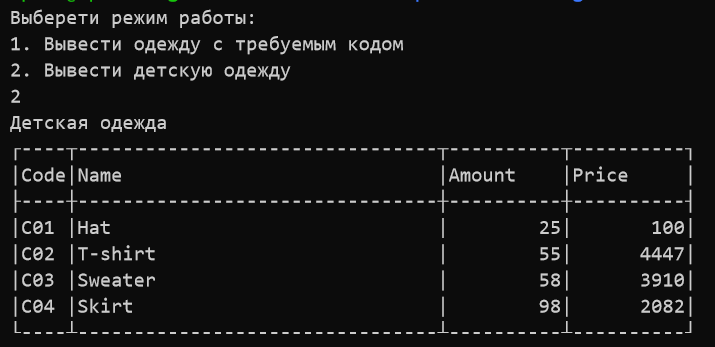
#include "module.h"  
  
using namespace std;  
  
DB edit\_DB(int raw, Item new\_item, DB data) {  
 data.table[raw] = new\_item;  
 return data;  
}

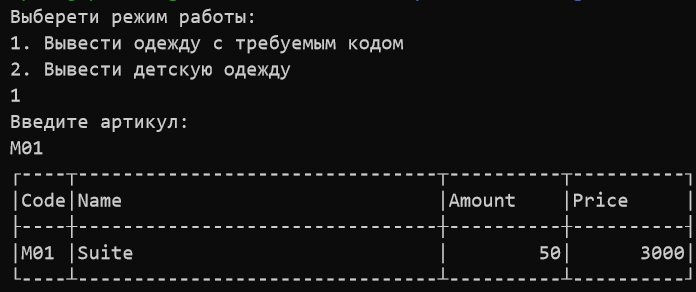
ОТЛАДКА И ТЕСТИРОВАНИЕ

Исходная таблица

W01,Dress,100,1000  
W02,T-shirt,7,4015  
W03,Blouse,16,357  
W04,Sweater,16,635  
W05,Skirt,74,3038  
W06,Jeans,57,3198  
M01,Suite,50,3000  
M02,T-shirt,40,587  
M03,Sweater,21,4604  
M04,Tracksuit,35,1447  
M05,Jeans,33,1094  
C01,Hat,25,100  
C02,T-shirt,55,4447  
C03,Sweater,58,3910  
C04,Skirt,98,2082

Вывод программы





Анализ результатов

Данный программный код отвечает всем требованиям проблемной части задачи.